APRB

AF IZW

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Burns et al.

Serial No.:	09/849,307	Group Art Unit:	2112
Filed:	5/7/2001	Examiner:	Clifford H. Knoll

Title: *A Producer/Consumer Locking System for Efficient Replication of File Data*

REPLY

MS Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In response to the Notification of Non-Compliant Appeal Brief mailed 11/30/2004, Applicant has attached a revised brief correcting all defects as described in the notification.

Respectfully submitted,

Ramraj Soundararaja

Ramraj Soundararajan
Registration No. 53,832

Lacasse & Associates, LLC
1725 Duke Street
Suite 650
Alexandria, Virginia 22314
(703) 838-7683
December 30, 2004

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPEAL BRIEF – 37 C.F.R § 1.192

U.S. Patent Application 09/849,307 entitled,

"A Producer/Consumer Locking System for Efficient Replication of File Data"

REAL PARTY OF INTEREST: International Business Machines Corporation

RELATED APPEALS AND INTERFERENCES:

None

STATUS OF CLAIMS:

Claims 1-12, 14-45, 58, and 61-67 are pending.

Claims 1-3, 5, 7-10, 14, 16-18, 20, 22-25, 28, 30-34, 36, 38-41, 58, 61-62, 64, and 66-67 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Leach (Common internet file system (CIFS/1.0) protocol: preliminary draft) in view of Miloushev (US 2002/0120763).

Claims 4, 6, 11-12, 15, 19, 21, 26-27, 29, 35, 37, and 42-45 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Leach in view of Bourne (US 2003/0120875).

Claims 63 and 65 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Leach and Miloushev as applied supra, further in view of Bourne.

Claims 1-12, 14-45, 58, and 61-67 are being appealed.

STATUS OF AMENDMENTS:

No amendments were filed after the final rejection of 05/17/2004.

SUMMARY OF CLAIMED SUBJECT MATTER:

(NOTE: All citations are made from the original specification, including the figures)

1. A locking system **(see figure 3 and 4)** implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system, said lock system comprising:

a consumer lock, said consumer lock granted to one or more readers **(see page 20, lines 15-16, and table on page 20)** and said consumer lock allowing a reader granted said consumer lock to read a file comprising one or more blocks of data;

a producer lock, said producer lock granted to a single writer **(see page 20, lines 15-16, and table on page 20)** and said producer lock allowing said writer granted said producer lock to update said file comprising one or more blocks of data, and wherein upon completion of said update , said writer releases said producer lock, and upon release of said producer lock, said updated file being published, with readers having a consumer lock associated with said updated file being notified regarding said update **(see page 21, lines 6-10)**.

2. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 1, wherein said file is updated by writing changed blocks of data to a physical storage location different than where said block of data is stored **(see figure 4; page 21, lines 4-5; page 22, line 1)**.

3. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 2, wherein, after said publication of said file, said system notifies readers granted a consumer lock for said file regarding location of said updated file **(see figure 4; page 22, lines 3-4)**.

4. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 2, wherein a copy of said file is held in a cache of said reader and changed blocks in said physical storage are updated in said cached copy, thereby providing updates at a finer granularity **(see page 21, lines 11-17; page 22, lines 19-20)**.

5. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 2, wherein reads performed on said block of data by said reader after receiving said notification are performed by reading said updated file at said notified location **(see page 21, lines 6-11)**.

6. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 2, wherein said reader continues to read said file from the physical storage location while said writer is writing updated data to said different physical storage location **(see page 21, lines 3-5)**.

7. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 1, wherein said writer writes data to storage devices physically separated from a storage device located on said file system server **(see page 21, lines 4-5)**.

8. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 7, wherein said writer writes data to said physically separate storage devices that are part of a storage area network **(see page 20, line 11 – page 21, line 17)**.

9. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 7, wherein said storage device located on said file system server stores metadata **(see page 13, lines 10-12)**.

10. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 7, wherein said physically separate storage devices cache data for read operations **(see page 21, lines 1-2)**.

11. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 1, wherein said reader is a web server **(see page 7, lines 15-19)**.

12. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 1, wherein said writer is a database management system **(see figure 4, DBMS)**.

Claim 13 (cancelled)

14. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 1, wherein said lock system is implemented on a system where said reader and said writer access data directly from storage devices via a storage area network and said readers and said writers access metadata from said file server via a data network separate from said storage area network **(see page 13, lines 10-12)**.

15. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 1, wherein said lock system is implemented in a distributed file system which utilizes multiple locking systems for data where the locking system used for a particular block of data is dependent on what application utilizes said particular block of data and the locking system utilized for the particular block of data is indicated by the metadata corresponding to said particular block of data **(see page 13, lines 10-12)**.

16. A method **(see figure 3, figure 4, and page 21, line 18 – page 22, line 6)** of updating a file comprising one or more data blocks in a distributed file system including a consumer lock **(see page 20, lines 15-16, and table on page 20)**, said consumer lock granted to multiple readers to allow said readers to read said file , and a producer lock **(see page 20, lines 15-16, and table on page 20)**, said producer lock granted to a single writer to allow said writer to update said file, said method comprising:

receiving a request from a writer to grant an exclusive producer lock **(see figure 4, element 402 and page 21, lines 19-20)**;

granting said producer lock to said writer**(see figure 4, element 402 and page 21, lines 19-20)**;

receiving a producer lock release message, said producer lock release message being received after said writer completes updating said file **(see page 22, lines 2-4)**; and

publishing said updated file **(see figure 4, element 408 and page 22, lines 1-3)**and sending an update message to said readers holding said consumer lock, said update message notifying said readers regarding said update **(see figure 4, elements 412, 414 and page 21, lines 3-5)**.

17. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a single writer to allow said writer to update said file as per claim 16, wherein said file is updated by writing changed blocks of data to a different physical storage location than where said data block is stored **(see figure 4; page 21, lines 4-5; page 22, line 1)**.

18. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file , and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 17, wherein said update message informs said readers granted a consumer lock for said file regarding location of said updated file **(see figure 4; page 22, lines 3-**

4).

19. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file , and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 17, wherein said update message causes a cached copy of said file held in a cache of said readers and changed blocks in said physical storage are updated in said cached copy, thereby providing updates at a finer granularity **(see page 21, lines 11-17; page 22, lines 19-20)**.

20. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 17, wherein reads performed on said data block by said readers after receiving said update message are performed by reading said updated file at said notified location **(see page 21, lines 6-11)**.

21. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 17, wherein said reader continues to read said file from the physical storage location while said writer is writing said updated file to said different physical storage location **(see page 21, lines 3-5)**.

22. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 16, wherein said writer writes data to storage devices physically separated from a storage device located on said file system server **(see page 21, lines 4-5)**.

23. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 22, wherein said writer writes data to said physically separate storage devices that are part of a storage area network **(see page 20, line 11 – page 21, line 17)**.

24. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 22, wherein said storage device located on said file system server stores metadata **(see page 13, lines 10-12)**.

25. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said data block file as per claim 22, wherein said physically separate storage devices cache data for read operations **(see page 21, lines 1-2)**.

26. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said data block file as per claim 16, wherein said reader is a web server **(see page 7, lines 15-19)**.

27. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 16, wherein said writer is a database management system **(see**

figure 4, DBMS).

28. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file , and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 16, wherein said lock system is implemented on a system where said readers and said writer access data directly from storage devices via a storage area network and said readers and said writers access metadata from said file server via a data network separate from said storage area network **(see page 13, lines 10-12).**

29. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file , and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 16, wherein said method is implemented in a distributed file system which utilizes multiple locking systems for data where the locking system used for a particular block of data is dependent on what application utilizes said particular block of data and the locking system utilized for the particular block of data is indicated by the metadata corresponding to said particular block of data **(see page 13, lines 10-12).**

30. A method **(see figure 3, figure 4 and page 21, line 18 – page 22, line 6)** of updating a file comprising one or more data blocks in a distributed file system including a consumer lock **(see page 20, lines 15-16, and table on page 20)**, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock **(see page 20, lines 15-16, and table on page 20)**, said producer lock granted to a writer to allow said writer to update said file, said method comprising:

sending a request for said producer lock **(see figure 4, 400)**;

receiving said producer lock **(see figure 4, 402)**;

updating said file comprising one or more data blocks **(see figure 4, 406)**;

releasing said producer lock after said updating is completed **(see figure 4, 408)**;

and

publishing said updated file **(see page 22, lines 2-3)**.

31. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 30, said method further comprising:

sending an update message to said readers granted said consumer lock after said releasing publishing step, said update message notifying said readers said file has been updated **(see page 21, lines 6-10)**.

32. A method of updating a data block file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said data block file as per claim 31, wherein said updating step comprises writing updated changed blocks of data to a different physical storage location than where said data block is stored **(see figure 4; page 21, lines 4-5; page 22, line 1)**.

33. (cancelled)

34. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said data block file as per claim 31, wherein said notification update message informs said readers granted a consumer lock for said file regarding location of said updated file **(see figure 4; page 22, lines 3-4)**.

35. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers

to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said data block file as per claim 32, wherein said update message causes a cached copy of said data block held in a cache of said readers and changed blocks in said physical storage are updated in said cached copy, thereby providing updates at a finer granularity **(see page 21, lines 11-17; page 22, lines 19-20)**.

36. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 34, wherein reads performed on said data block by said readers after receiving said update message are performed by reading said updated file from said notified location **(see page 21, lines 6-11)**.

37. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 32, wherein said readers continue to read said file from the physical storage location while said writer is writing updated data to said different physical storage location **(see page 21, lines 3-5)**.

38. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 30, wherein said writer writes data to storage devices physically separated from a storage device located on said file system server **(see page 21, lines 4-5)**.

39. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer

to update said file, as per claim 38, wherein said writer writes data to said physically separate storage devices that are part of a storage area network **(see page 20, line 11 – page 21, line 17)**.

40. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 30, wherein said storage device located on said file system server stores metadata **(see page 13, lines 10-12)**.

41. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 38, wherein said physically separate storage devices cache data for read operations **(see page 21, lines 1-2)**.

42. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 30, wherein said reader is a web server **(see page 7, lines 15-19)**.

43. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 30, wherein said writer is a database management system **(see figure 4, DBMS)**.

44. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer

to update said file as per claim 30, wherein said method is implemented on a system where said readers and said writer access data directly from storage devices via a storage area network and said readers and said writers access metadata from said file server via a data network separate from said storage area network **(see page 13, lines 10-12)**.

45. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 30, wherein said method is implemented in a distributed file system which utilizes multiple locking systems for data where the locking system used for a particular block of data is dependent on what application utilizes said particular block of data and the locking system utilized for the particular block of data is indicated by the metadata corresponding to said particular block of data **(see page 13, lines 10-12)**.

46 - 57. (cancelled)

58. A distributed computing system **(see figure 3)** including a file system handling cache coherency and data consistency providing quality of service through a locking protocol, said system comprising:

a server **(see figure 3, element 300)**, said server connected to at least one client **(see figure 3, element 306)** of said distributed computing system via a first data network **(see figure 3, element 304)**, said server serving file metadata to said client upon said client accessing a file stored in said distributed computing system, said server managing data consistency and cache coherency through said locking protocol;

a storage device **(see figure 3, element 302)**connected to said client via a second data network, said storage device storing file data;

wherein one of said locking protocol comprises the following locks:

a consumer lock **(see page 20, lines 15-16, and table on page 20)**, said consumer lock granted to one or more readers and said consumer lock allowing a reader granted said consumer lock to read a file comprising one or more blocks of data; and

a producer lock **(see page 20, lines 15-16, and table on page 20)**, said producer lock granted to a single writer and said producer lock allowing said writer granted said producer lock to update said file comprising one or more blocks of data, and upon completion of said update, said writer releases said producer lock, and upon release of said producer lock, said updated file being published, with readers having a consumer lock associated with said updated file being notified regarding said update.

59. (cancelled)

60. (cancelled)

61. A distributed computing system including a file system handling cache coherency and data consistency providing quality of service through a locking protocol, as per claim 58, wherein said file is changed by writing changed blocks of data to a physical storage location different than where said block of data is stored **(see figure 4, page 21, lines 4-5; page 22, line 1)**.

62. A distributed computing system including a file system handling cache coherency and data consistency providing quality of service through a locking protocol, as per claim 61, wherein, after said publication of said file, said system notifies readers granted a consumer lock for said file regarding location of said updated file **(see figure 4; page 22, lines 3-4)**.

63. A distributed computing system including a file system handling cache coherency and data consistency providing quality of service through a locking protocol, as per claim 62, wherein a cached copy of said file held in a cache of said reader and changed blocks in said physical storage are updated in said cached copy, thereby providing updates at a finer granularity **(see page 21, lines 11-17; page 22, lines 19-20)**.

64. A distributed computing system including a file system handling cache coherency and data

consistency providing quality of service through a locking protocol, as per claim 62, wherein reads performed on said file are performed by reading updated data from said notified location **(see page 21, lines 6-11)**.

65. A distributed computing system including a file system handling cache coherency and data consistency providing quality of service through a locking protocol, as per claim 61, wherein said reader continues to read said file from the physical storage location while said writer is writing updated file to said different physical storage location **(see page 21, lines 3-5)**.

66. A distributed computing system including a file system handling cache coherency and data consistency providing quality of service through a locking protocol, as per claim 58, wherein said reader is a web server **(see page 7, lines 15-19)**.

67. A distributed computing system including a file system handling cache coherency and data consistency providing quality of service through a locking protocol, as per claim 58, wherein said writer is a database management system **(see figure 4, DBMS)**.

68. (cancelled)

GROUNDS OF REJECTIONS TO BE REVIEWED ON APPEAL:

1. Was a proper rejection made under existing USPTO guidelines with respect to claims 1-3, 5, 7-10, 14, 16-18, 20, 22-25, 28, 30-34, 36, 38-41, 58, 61-62, 64, and 66-67, which stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Leach (Common internet file system (CIFS/1.0) protocol: preliminary draft) in view of Miloushev (US 2002/0120763)?

2. Was a proper rejection made under existing USPTO guidelines with respect to claims 4, 6, 11-12, 15, 19, 21, 26-27, 29, 35, 37, and 42-45, which stand rejected under 35 U.S. C. § 103(a) as being unpatentable over Leach in view of Bourne (US 2003/0120875)?

3. Was a proper rejection made under existing USPTO guidelines with respect to claims 63 and 65, which stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Leach and Miloushev as applied supra, further in view of Bourne.

ARGUMENT:

I. *Arguments for rejections with respect to claims 1-3, 5, 7-10, 14, 16-18, 20, 22-25, 28, 30-34, 36, 38-41, 58, 61-62, 64, and 66-67, which stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Leach (Common internet file system (CIFS/1.0) protocol: preliminary draft) in view of Miloushev (US 2002/0120763)*

To establish a prima facie case of obviousness under U.S.C. § 103, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. Additionally, the teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and should not be based on applicant's disclosure (In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991)). Applicants contend, as will be shown below, that the rejections of claims 28, 30-34, 36, 38-41, 58, 61-62, 64, and 66-67 are improper as they fail to meet many of the criteria listed above.

Independent claim 1 and 16 teach a locking system and method that is implemented in a distributed file system where clients directly access data on storage devices on a storage area network. Both claims 1 and 16 teach a consumer lock that is granted to one or more readers and a producer lock granted to a single writer to update a file comprising one or more blocks of data. Claims 1 and 16 also teach that, upon completion of an update, the writer releases the producer lock, and upon releasing the producer lock, the updated file is published, with readers having a consumer lock (associated with the file) being notified of the update.

On page 2 of the office action, the examiner equates the consumer lock of applicants' invention to the "level II oplock" described in §2.7.3 of the Leach reference (see page 15 and 16 of Leach). However, a closer reading of the citation and the Leach reference in its entirety show otherwise. For example, on page 16 in §2.7.3 of the Leach reference it states that "Level II oplocks allow multiple clients to have the same file open, providing that no client is performing a write operation to the file." Additionally, on page 13, §2.7 of the Leach reference, it states that "A Level II oplock indicates that there are multiple readers of a file and no writers." This teaches away from applicants' invention which teaches that a single writer can hold a producer lock while the readers have a consumer lock (see claim 1: "with readers having a consumer lock...being notified of the update" and claim 16: "after said writer completes updating said file...sending an update message to said readers holding said consumer lock"). Therefore, the Leach reference does not read on or describe such locks.

The examiner also equates the producer lock of applicants' invention with the "exclusive oplocks" described in §2.7.1 of the Leach reference (see pages 13 and 14 of the Leach reference). However, a closer reading of the citation and the reference in its entirety shows otherwise. For example, in §2.7.1 Leach states that "if the file is open by anyone else, the client is refused the oplock." As mentioned above, this teaches away from applicants' invention which teaches that a single writer can hold the producer lock while multiple readers have a consumer lock. Therefore, the Leach reference does not read on or describe such a limitation.

Furthermore, claims 1 and 16 also teach that upon completion of an update (by a writer holding the producer lock), the writer releases the producer lock after which the updated file is published, with readers having a consumer lock being notified of the update. Applicants' contend that the examiner builds on previous erroneous statements which equates the producer and consumer lock to "exclusive oplock" and "level II oplock" and further states that the Leach reference in §2.7.3 and §1.1.4 teaches "if any write operation is performed it need only notify the level II clients." Applicants, however, contend that the examiner erroneously equates partial citations of §2.7.3 and §1.1.4 with the above-mentioned limitation of claims 1 and 16.

Specifically, applicants direct attention to the entire sentence (which was omitted by the examiner as he only includes a partial recitation) in §2.7.3 which states that "This allows the server to guarantee that if any write operation is performed, it need only notify the level II clients that the lock should be broken without having to synchronize all of the accessors of the file." This recitation states that if a write operation is to be performed while level II clients are holding a read lock, the Leach system notifies all clients that the level II read locks need to be broken prior to granting access to a writer, a limitation that teaches away from the applicants' invention that teaches that a single writer can hold the producer lock while multiple readers have a consumer lock (i.e., the read lock need not be broken).

Additionally, the recitation of §2.7.3 states that the level II lock is broken without having to synchronize all of the accessors (readers) of the file, which further reinforces the point that Leach teaches breaking the level II read lock when a writer wants to modify a file (with readers not having any access to the file). With applicants' invention, on the other hand, readers with a consumer lock, C, are able to still read the file while a writer with a producer lock, P, is modifying it.

Furthermore, the examiner also states that Leach does not expressly mention that the producer releases the lock, and further states that "this is manifestly the obvious use of a lock, as exemplified by Miloushev." In support of his argument, the examiner states in page 3 of the office action, that the Miloushev reference, in paragraph 230, discloses a writer releasing a lock. Applicants', however, contend that such a limitation is neither manifestly obvious nor specifically shown in the Miloushev reference. Specifically, the Miloushev reference merely states, in paragraph 230, that the client writes to a disk (which is a mirror disk) and "then unlocks the region to complete the write transaction." Miloushev, however, fails to either explicitly or implicitly disclose a system or method wherein, upon completion of an update, and followed by the release of the producer lock (granted to one user), the updated file is published (only after the release of the producer lock), wherein readers having a consumer lock are subsequently notified of the update. Hence, applicants strongly disagree with the notion that it is manifestly obvious to have released a producer lock followed by the publication of the updated file, wherein, after publication, readers (holding consumer locks for that particular file) are notified of the update. Therefore, applicants contend that the Leach reference does not read on or describe such a limitation.

With regard to claims 30-32, the examiner repeats the recitations with respect to independent claims 1 and 16. The arguments presented above substantially apply to claims 30-32. Similarly, the examiner has repeated the recitations with respect to claims and 16 for independent claim 58 and, hence, the arguments presented above substantially apply to claim 58.

With regard to claims 2 and 17, the examiner states that the limitation wherein a "file is updated by writing changed blocks of data to a physical storage location different than where said block of data is stored" is taught in §1.1.3 entitled "Safe caching, read ahead, and write-behind." A closer reading of the cited paragraph merely states that files are cached by the reader or writer, a concept that is well known in the art. For example, page 9, lines 3-10 of the application-as-filed is reproduced below for illustrating prior art caching solutions including some of their disadvantages:

> *"The disadvantage of AFS is that it does not correctly implement an efficient model for data replication. The actual behavior is that the AFS clients write dirty data back to the server when closing a file, and AFS servers send callback invalidation messages whenever clients write data. In most cases, these policies result in an appropriate consistency. <u>However, if a writing client writes back some portion of its cache without closing the file, a callback is sent to all registered clients, and reading clients can see partial updates. This most often occurs when a writing client, in our example of the DBMS, operates under a heavy load or on large files. In these cases, the cache manager writes back dirty blocks to the server to reclaim space</u>."*

It can be seen from the application-as-filed that a problem with caching solutions is that, under a heavy load or on large files, the cache manager (which is limited in size) writes back dirty blocks (i.e., a dirty write) to the server holding the original data so that it can reclaim space for further operations on the file. Hence, it is clear that in caching solutions, when a need arises for more space, a dirty write is performed. Claim 2 and 17, on the other hand, builds on the limitations of independent claims 1 and 16, and further adds the limitation of an out-of-place write, wherein the changed blocks are written to a physical storage location in a storage area network (not a cache) that is different that where the data is stored. This eliminates the dirty write problem associated with caching systems. Applicants contend that the recitation relied on by the examiner clearly states that the system uses a cache and is silent about out-of-place writes to a physical storage location in a storage area network.

Regarding claims 3 (which depends on claim 2) and claim 18 (which depends on claim 17), the examiner states that the previously cited limitations of §1.1.3 ("read caching") as support for his rejection. Applicants wish to state that the arguments presented above with respect to claims 1-2 and 16-17 substantially apply to claims 3 and 18 respectively as they inherit the limitations of the claims from which they depend. Regarding claims 5 (which depends on claim 2), claim 20 (which depends on claim 17), and claim 36 (which depends on claim 34, which further depends on claim 31), the examiner cites the limitations described in §1.1.4 as support for his rejection. Similarly, the arguments presented above with respect to claims 1-2, 16-17, and 30-31 substantially apply to claims 5, claim 20 and claim 36 respectively, as they inherit the limitations of the claims from which they depend. Additionally, applicants wish to emphasize that the Leach reference fails to teach a method and system based on a producer and a consumer lock that performs out-place writes and, upon publication of an updated file, notifies readers regarding the location of the updated file.

With respect to claims 7-10, 22-25, and 38-41, the examiner states that "Leach fails to disclose physically separate block for writing", but states that Miloushev, in paragraph 414, discloses "writing data to storage devices physically separate from the storage device located on said file server." But a closer reading of the citation, and the Mikloushev reference in its entirety, merely suggests that updates to a file are stored in a cache via "client-side caching" (see paragraph 414 and 415). This solution suffers from the previously described caching problem, wherein, under a heavy load or on large files, the cache manager (which is limited in size) writes back dirty blocks (i.e., a dirty write) to the server holding the original data so that it can reclaim space for further operations on the file. Hence, in caching solutions, when a need arises for more space, a dirty write is performed; an implementation that teaches away from' the present invention which teaches an out-of-place write of update data and which allows publication of a file after the publisher lock is released. Hence, applicants contend that Miloushev in combination with Leach fail to address such an out-of-place write, and applicants also contend that the examiner has erroneously equated the out-of-place write with prior art caching solutions.

Regarding claims 14, 28, and 44, applicants agree with the examiner that the limitations of these claims are not taught by the Leach reference. However, applicants disagree with the examiner that such limitations are disclosed in the Miloushev reference. For support, the examiner relies on paragraph 115 and 269 of the Miloushev reference. A closer reading of the citations, however, merely mention problems with "arbitration" in a multi-client/multi-server system (see paragraph 115) and a "spillover" mechanism (see paragraph 269). The citations, however, fail to teach a system and method that use publisher (granted to one writer) and consumer locks (granted to many readers), wherein readers with the consumer lock are able to access data directly from storage devices in a SAN and the writer with a publisher lock is able to access metadata from a file server over a data network separate from the SAN.

The arguments presented above with respect to independent claim 58 substantially apply to dependent claims 61, 62, 64, 66, and 67, as they inherit the limitations of the claim from which they depend. Specifically, applicants emphasize that neither the Leach reference nor the Miloushev reference teach the maintenance of a producer lock for writing data and consumer locks for reading data, wherein the readers holding the consumer lock associated with the file being updated are notified of such an update after the file publishes. Also, neither the Leach reference nor the Miloushev reference teach an out-of-place write and applicants contend that the citations merely teach caching data, which is erroneously equated by the examiner to an out-of-place write.

Hence, applicants contend that the examiner has failed to establish a prima facie case of obviousness under U.S.C. § 103, as there is no suggestion or motivation, either in the cited references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings.

II. *Arguments for rejections with respect to claims 4, 6, 11-12, 15, 19, 21, 26-27, 29, 35, 37, and 42-45, which stand rejected under 35 U.S. C. § 103(a) as being unpatentable over Leach in view of Bourne (US 2003/0120875)*

Regarding claims 4, 19, and 35, the examiner states that Leach does not expressly mention updating changed blocks of data in cache. The examiner further states that the Bourne reference, in paragraph 86, discloses updating changed blocks of data at a finer granularity. A closer reading of the citation and the Bourne reference in its entirety merely suggests a "fragment granularity". But, a closer read of paragraph 86 of the Bourne reference merely suggests a "fragment granularity" which is defined as "whole pages, also referred to as fragments". By stark contrast, applicants' invention updates blocks of data (not whole web pages) at a finer granularity. Additionally, there is no teaching or suggestion in the Bourne reference for implementing Bourne's system/method with locks such as consumer or producer locks. Hence, there is no explicit or implicit suggestion in the Bourne reference for updating data at a finer granularity.

Regarding claims 6, 21, and 37, the examiner states that Leach does not expressly disclose a reader "that continuous to read data". He further cites paragraph 53 of the Bourne reference as providing such a limitation. A closer reading of paragraph 53 merely suggests that "dynamic" content in a webpage is avoided. For example, if the system/method of Bourne identifies that a webpage to be loaded has dynamic content. Then, that particular dynamic content is avoided (i.e., not retrieved). This is in contrast to the present invention that provides a producer lock for writing data and consumer locks for reading data, wherein the readers holding the consumer lock associated with the file being updated are notified of such an update after the file publishes.

The arguments presented above with respect to independent claim 1, 16, 30, and 58 substantially apply to dependent claims 11-12, 26-27, 42-43 as they inherit the limitations of the claim from which they depend.

Regarding claims 15, 29, and 45, the examiner contends that the Bourne reference discloses multiple locking systems for data, wherein the locking system used for a particular block is dependent on what application utilizes the particular block of data and the locking system is indicated by the metadata. In support of this argument, the examiner has cited paragraph 84 in page 7 of the office action. A closer reading of the citation and the reference in its entirety merely reveals a discussion of figure 11 which includes an example of a Java Server Page (JSP). Notably absent is any explicit or implicit mention of locking system. Also, absent in the citations is a locking system or a locking system that is indicated in the metadata.

Hence, applicant contends that the examiner has failed to establish a prima facie case of obviousness under U.S.C. § 103, as there is no suggestion or motivation, either in the cited references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings.

III. *Arguments for rejections with respect to claims 63 and 65, which stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Leach and Miloushev as applied supra, further in view of Bourne*

Regarding claim 63 and 65, the examiner reiterates that the Bourne reference, in paragraph 86, discloses updating changed blocks of data at a finer granularity. A closer reading of the citation and the Bourne reference in its entirety merely suggests a "fragment granularity". But, a closer read of paragraph 86 of the Bourne reference merely suggests a "fragment granularity" which is defined as "whole pages, also referred to as fragments can be cached". As mentioned above, applicants' invention updates blocks of data (not whole web pages) at a finer granularity. Additionally, there is no teaching or suggestion in the Bourne reference for implementing Bourne's system/method with locks such as consumer or producer locks. There is also no suggestion in the Bourne reference for updating data at a finer granularity.

Hence, applicant contends that the examiner has failed to establish a prima facie case of obviousness under U.S.C. § 103, as there is no suggestion or motivation, either in the cited references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings.

CLAIMS APPENDIX:

1. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system, said lock system comprising:

a consumer lock, said consumer lock granted to one or more readers and said consumer lock allowing a reader granted said consumer lock to read a file comprising one or more blocks of data;

a producer lock, said producer lock granted to a single writer and said producer lock allowing said writer granted said producer lock to update said file comprising one or more blocks of data, and

wherein upon completion of said update , said writer releases said producer lock, and upon release of said producer lock, said updated file being published, with readers having a consumer lock associated with said updated file being notified regarding said update.

2. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 1, wherein said file is updated by writing changed blocks of data to a physical storage location different than where said block of data is stored.

3. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 2, wherein, after said publication of said file, said system notifies readers granted a consumer lock for said file regarding location of said updated file.

4. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 2, wherein a copy

of said file is held in a cache of said reader and changed blocks in said physical storage are updated in said cached copy, thereby providing updates at a finer granularity.

5. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 2, wherein reads performed on said block of data by said reader after receiving said notification are performed by reading said updated file at said notified location.

6. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 2, wherein said reader continues to read said file from the physical storage location while said writer is writing updated data to said different physical storage location.

7. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 1, wherein said writer writes data to storage devices physically separated from a storage device located on said file system server.

8. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 7, wherein said writer writes data to said physically separate storage devices that are part of a storage area network.

9. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data

and manages revocation and granting of locks of said lock system as per claim 7, wherein said storage device located on said file system server stores metadata.

10. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 7, wherein said physically separate storage devices cache data for read operations.

11. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 1, wherein said reader is a web server.

12. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 1, wherein said writer is a database management system.

Claim 13 (cancelled)

14. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data and manages revocation and granting of locks of said lock system as per claim 1, wherein said lock system is implemented on a system where said reader and said writer access data directly from storage devices via a storage area network and said readers and said writers access metadata from said file server via a data network separate from said storage area network.

15. A locking system implemented on a distributed file system where clients directly access data on storage devices via a storage area network and a file server provides metadata for said data

and manages revocation and granting of locks of said lock system as per claim 1, wherein said lock system is implemented in a distributed file system which utilizes multiple locking systems for data where the locking system used for a particular block of data is dependent on what application utilizes said particular block of data and the locking system utilized for the particular block of data is indicated by the metadata corresponding to said particular block of data.

16. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file , and a producer lock, said producer lock granted to a single writer to allow said writer to update said file, said method comprising:

receiving a request from a writer to grant an exclusive producer lock;

granting said producer lock to said writer;

receiving a producer lock release message, said producer lock release message being received after said writer completes updating said file; and

publishing said updated file and sending an update message to said readers holding said consumer lock, said update message notifying said readers regarding said update.

17. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file , and a producer lock, said producer lock granted to a single writer to allow said writer to update said file as per claim 16, wherein said file is updated by writing changed blocks of data to a different physical storage location than where said data block is stored.

18. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file , and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 17, wherein said update message informs said readers granted a consumer lock for said file regarding location of said updated file.

19. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file , and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 17, wherein said update message causes a cached copy of said file held in a cache of said readers and changed blocks in said physical storage are updated in said cached copy, thereby providing updates at a finer granularity.

20. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file , and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 17, wherein reads performed on said data block by said readers after receiving said update message are performed by reading said updated file at said notified location.

21. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 17, wherein said reader continues to read said file from the physical storage location while said writer is writing said updated file to said different physical storage location.

22. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file , and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 16, wherein said writer writes data to storage devices physically separated from a storage device located on said file system server.

23. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers

to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 22, wherein said writer writes data to said physically separate storage devices that are part of a storage area network.

24. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file , and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 22, wherein said storage device located on said file system server stores metadata.

25. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file , and a producer lock, said producer lock granted to a writer to allow said writer to update said data block file as per claim 22, wherein said physically separate storage devices cache data for read operations.

26. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file , and a producer lock, said producer lock granted to a writer to allow said writer to update said data block file as per claim 16, wherein said reader is a web server.

27. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file , and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 16, wherein said writer is a database management system.

28. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file , and a producer lock, said producer lock granted to a writer to allow said writer

to update said file as per claim 16, wherein said lock system is implemented on a system where said readers and said writer access data directly from storage devices via a storage area network and said readers and said writers access metadata from said file server via a data network separate from said storage area network.

29. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file , and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 16, wherein said method is implemented in a distributed file system which utilizes multiple locking systems for data where the locking system used for a particular block of data is dependent on what application utilizes said particular block of data and the locking system utilized for the particular block of data is indicated by the metadata corresponding to said particular block of data.

30. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file, said method comprising:

sending a request for said producer lock;

receiving said producer lock;

updating said file comprising one or more data blocks;

releasing said producer lock after said updating is completed; and

publishing said updated file.

31. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 30, said method further comprising:

sending an update message to said readers granted said consumer lock after said releasing publishing step, said update message notifying said readers said file has been updated.

32. A method of updating a data block file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said data block file as per claim 31, wherein said updating step comprises writing updated changed blocks of data to a different physical storage location than where said data block is stored.

33. (cancelled)

34. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said data block file as per claim 31, wherein said notification update message informs said readers granted a consumer lock for said file regarding location of said updated file .

35. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said data block file as per claim 32, wherein said update message causes a cached copy of said data block held in a cache of said readers and changed blocks in said physical storage are updated in said cached copy, thereby providing updates at a finer granularity.

36. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 34, wherein reads performed on said data block by said readers after receiving said update message are performed by reading said updated file from said notified

location.

37. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 32, wherein said readers continue to read said file from the physical storage location while said writer is writing updated data to said different physical storage location.

38. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 30, wherein said writer writes data to storage devices physically separated from a storage device located on said file system server.

39. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file, as per claim 38, wherein said writer writes data to said physically separate storage devices that are part of a storage area network.

40. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 30, wherein said storage device located on said file system server stores metadata.

41. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers

to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 38, wherein said physically separate storage devices cache data for read operations.

42. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 30, wherein said reader is a web server.

43. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 30, wherein said writer is a database management system.

44. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 30, wherein said method is implemented on a system where said readers and said writer access data directly from storage devices via a storage area network and said readers and said writers access metadata from said file server via a data network separate from said storage area network.

45. A method of updating a file comprising one or more data blocks in a distributed file system including a consumer lock, said consumer lock granted to multiple readers to allow said readers to read said file, and a producer lock, said producer lock granted to a writer to allow said writer to update said file as per claim 30, wherein said method is implemented in a distributed file system which utilizes multiple locking systems for data where the locking system used for a particular block of data is dependent on what application utilizes said particular block of data and the locking system utilized for the particular block of data is indicated by the metadata

corresponding to said particular block of data.

46 - 57. (cancelled)

58. A distributed computing system including a file system handling cache coherency and data consistency providing quality of service through a locking protocol, said system comprising:

a server, said server connected to at least one client of said distributed computing system via a first data network, said server serving file metadata to said client upon said client accessing a file stored in said distributed computing system, said server managing data consistency and cache coherency through said locking protocol ;

a storage device connected to said client via a second data network, said storage device storing file data;

wherein one of said locking protocol comprises the following locks:

a consumer lock, said consumer lock granted to one or more readers and said consumer lock allowing a reader granted said consumer lock to read a file comprising one or more blocks of data; and

a producer lock, said producer lock granted to a single writer and said producer lock allowing said writer granted said producer lock to update said file comprising one or more blocks of data, and upon completion of said update, said writer releases said producer lock, and upon release of said producer lock, said updated file being published, with readers having a consumer lock associated with said updated file being notified regarding said update.

59. (cancelled)

60. (cancelled)

61. A distributed computing system including a file system handling cache coherency and data consistency providing quality of service through a locking protocol, as per claim 58, wherein said file is changed by writing changed blocks of data to a physical storage location different

than where said block of data is stored.

62. A distributed computing system including a file system handling cache coherency and data consistency providing quality of service through a locking protocol, as per claim 61, wherein, after said publication of said file, said system notifies readers granted a consumer lock for said file regarding location of said updated file.

63. A distributed computing system including a file system handling cache coherency and data consistency providing quality of service through a locking protocol, as per claim 62, wherein a cached copy of said file held in a cache of said reader and changed blocks in said physical storage are updated in said cached copy, thereby providing updates at a finer granularity.

64. A distributed computing system including a file system handling cache coherency and data consistency providing quality of service through a locking protocol, as per claim 62, wherein reads performed on said file are performed by reading updated data from said notified location.

65. A distributed computing system including a file system handling cache coherency and data consistency providing quality of service through a locking protocol, as per claim 61, wherein said reader continues to read said file from the physical storage location while said writer is writing updated file to said different physical storage location.

66. A distributed computing system including a file system handling cache coherency and data consistency providing quality of service through a locking protocol, as per claim 58, wherein said reader is a web server.

67. A distributed computing system including a file system handling cache coherency and data consistency providing quality of service through a locking protocol, as per claim 58, wherein said writer is a database management system.

68. (cancelled)

EVIDENCE APPENDIX:

The affidavit under 37 CFR 1.131 (executed by each inventor), which was previously submitted along with the response of 02/27/2004 has been included with the appeal brief.

ARC920000024US1
09/849,307

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Burns et al.

Serial No.:	09/849,307	Group Art Unit:	2189
Filed:	May 7, 2001	Examiner:	Clifford H. Knoll

Title: *A Producer/Consumer Locking System for Efficient Replication of File Data*

AFFIDAVIT UNDER 37 CFR 1.131

MS Non-Fee Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

As a below inventor, I hereby declare that:

1) I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the patent application filed May 7, 2001 for A Producer/Consumer Locking System for Efficient Replication of File Data and inventor of the subject matter described and claimed therein.

2) Prior to March 26, 2001, I conceived the invention as described and claimed in the subject application and as amended by the amendment accompanying this affidavit in the United States as evidence by DISCLOSURE MATERIAL, attached hereto as Exhibit A.

3) From prior to March 26, 2001, until the filing of the patent application on May 7, 2001, I exercised due diligence toward reducing the invention to practice, as evidenced by DISCLOSURE MATERIAL, attached hereto as Exhibit A. The disclosure was evaluated and processed as part of IBM's standard patent processing procedures and culminated in the filing of the patent application on May 7, 2001.

4) The photocopies of DISCLOSURE MATERIAL attached to this affidavit as Exhibit A are true copies of the original pages showing conception of the invention prior to March 26, 2001 coupled with due diligence from prior to March 26, 2001 to the filing of the patent application.

Date February 16, 2004 ______________________ L.S.
Randal Chilton Burns

State of Maryland ______________________

County of Baltimore ______________________

Sworn to and subscribed before me this 16th day of February, 2004.

Jean Kajder Pugh
Notary Public Jean Kajder Pugh

Date ____________, 2004 ______________________ L.S.
Robert Michael Rees

State of ______________________

County of ______________________

Sworn to and subscribed before me this ________ day of ____________, 2004.

____
Notary Public

Date ____________, 2004 ______________________ L.S.
Atul Goel

State of ______________________

County of ______________________

Sworn to and subscribed before me this ________ day of ____________, 2004.

____
Notary Public

Date ____________, 2004 ______________________ L.S.
Wayne Curtis Hineman

State of ______________________

County of ______________________

Sworn to and subscribed before me this ________ day of ____________, 2004.

____
Notary Public

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of: Burns et al.

Serial No.: 09/849,307 Group Art Unit: 2189

Filed: May 7, 2001 Examiner: Clifford H. Knoll

Title: *A Producer/Consumer Locking System for Efficient Replication of File Data*

AFFIDAVIT UNDER 37 CFR 1.131

MS Non-Fee Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

As a below inventor, I hereby declare that:

1) I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the patent application filed May 7, 2001 for A Producer/Consumer Locking System for Efficient Replication of File Data and inventor of the subject matter described and claimed therein.

2) Prior to March 26, 2001, I conceived the invention as described and claimed in the subject application and as amended by the amendment accompanying this affidavit in the United States as evidence by DISCLOSURE MATERIAL, attached hereto as Exhibit A.

3) From prior to March 26, 2001, until the filing of the patent application on May 7, 2001, I exercised due diligence toward reducing the invention to practice, as evidenced by DISCLOSURE MATERIAL, attached hereto as Exhibit A. The disclosure was evaluated and processed as part of IBM's standard patent processing procedures and culminated in the filing of the patent application on May 7, 2001.

4) The photocopies of DISCLOSURE MATERIAL attached to this affidavit as Exhibit A are true copies of the original pages showing conception of the invention prior to March 26, 2001 coupled with due diligence from prior to March 26, 2001 to the filing of the patent application.

Date____________, 2004 ______________________ L.S.
Randal Chilton Burns

State of ______________________

County of ______________________

Sworn to and subscribed before me this ________ day of ____________, 2004.

____
Notary Public

Date 2/23, 2004 ______________________ L.S.
Robert Michael Rees

State of California ______________________

County of Santa Clara ______________________

Sworn to and subscribed before me this 23rd day of February, 2004.

Errol O. Smedley
Notary Public

ERROL O. SMEDLEY
Comm. # 1304907
NOTARY PUBLIC - CALIFORNIA
Santa Clara County
My Comm. Expires May 18, 2005

Date____________, 2004 ______________________ L.S.
Atul Goel

State of ______________________

County of ______________________

Sworn to and subscribed before me this ________ day of ____________, 2004.

____
Notary Public

Date 2/23, 2004 Wayne C. Hineman L.S.
Wayne Curtis Hineman

State of California ______________________

County of Santa Clara ______________________

Sworn to and subscribed before me this 23rd day of February, 2004.

Errol O. Smedley
Notary Public

ERROL O. SMEDLEY
Comm. # 1304907
NOTARY PUBLIC - CALIFORNIA
Santa Clara County
My Comm. Expires May 18, 2005

Page 2 of 2

Date____________________, 2004 ____L.S.
Randal Chilton Burns

State of ____

County of ____

Sworn to and subscribed before me this ____________ day of ________________, 2004.

____
Notary Public

Date____________________, 2004 ____L.S.
Robert Michael Rees

State of ____

County of ____

Sworn to and subscribed before me this ____________ day of ________________, 2004.

____
Notary Public

Date 10th FEBRUARY, 2004 Atul Goel L.S.
Atul Goel

State of CALIFORNIA

County of SANTA CLARA

Sworn to and subscribed before me this 10th day of Feb, 2004.

Pavneet Singh.
Notary Public

Date____________________, 2004 ____L.S.
Wayne Curtis Hineman

State of ____

County of ____

Sworn to and subscribed before me this ____________ day of ________________, 2004.

____
Notary Public

As this Appeal Brief has been timely filed within the set period of response, no petition for extension of time or associated fee is required. However, the Commissioner is hereby authorized to charge any deficiencies in the fees provided, to include an extension of time, to Deposit Account No. 09-0441.

Respectfully submitted by
Applicant's Representative,

Ramraj Soundararajan

Ramraj Soundararajan
Reg. No. 53,832

1725 Duke Street
Suite 650
Alexandria, VA 22314
(703) 838-7683